A package for calculating elastic tensors of cubic phases using WIEN.

Thomas CHARPIN
Laboratoire des Géomatériaux de l'IPGP
4,pl Jussieu
F-75252 Paris, France

12th July 2001

The package *elast* provides a set of programs and scripts that allows elastic tensor calculations for cubic phases (primitive, body-centered, or face centered) by using WIEN. Given a starting cubic structure, the package

- generates WIEN input files simulating strained structures,
- generates scripts to make WIEN calculate these structures,
- analyze the results, plot them, and derive the linear elastic parameters.

1 Theoretical background and notations.

Let E^0_{tot} be the total energy of an initial crystal, and V_0 its volume. By bending this crystal, the energy, E_{tot}, of the resulting strained state can be expressed as:

$$E_{tot} = E^0_{tot} + P(V - V_0) + \phi_{elast}, \tag{1}$$

where V is the volume of the strained lattice, ϕ_{elast} the elastic energy, and P the pressure defined by:

$$P = -\left(\frac{\partial E^0_{tot}}{\partial V}\right)(V_0)$$

To first order, the strained lattice (lattice vectors $\overleftrightarrow{a}$) is related to the unstrained lattice ($\overleftrightarrow{a}_0$) by $\overleftrightarrow{a} = (\mathcal{I} + \overline{\epsilon}).\ \overleftrightarrow{a}_0$, where $\mathcal{I}$ is the identity matrix and $\overline{\epsilon}$ the strain tensor. According to Hooke's law, the linear elastic constants ($\mathcal{C}_{ijkl}$) are then defined by using the second order development of the elastic energy:

$$\phi_{elast} = \frac{V}{2}.\mathcal{C}_{ijkl}\epsilon_{ij}\epsilon_{kl} \qquad (i,j,k,l = 1,2,3)$$

or , in the Voigt's two-suffix notation:

$$\phi_{elast} = \frac{V}{2}.\mathcal{C}_{ij}\epsilon_i\epsilon_j \qquad (i,j = 1,2,3,4,5,6) \tag{2}$$

The $(V - V_0)$ term in equation (1) is linear with respect to strain: $V - V_0 = V_0.Tr(\overline{\epsilon}) = V_0.\sum_{i=1}^{3}\epsilon_i$. Thus, it is possible to derive elastic constants from the second-order derivatives of E_{tot}:

$$\mathcal{C}_{ij} = \frac{1}{V_0}.\frac{\partial^2 E_{tot}}{\partial\epsilon_i\partial\epsilon_j} \tag{3}$$

2 Implementation.

A cubic crystal has only three independent elastic constants, $\mathcal{C}_{11}$, $\mathcal{C}_{12}$, and $\mathcal{C}_{44}$ leading to an effective elastic tensor (in two-suffix notation):

$$\mathcal{C} = \begin{pmatrix} \mathcal{C}_{11} & \mathcal{C}_{12} & \mathcal{C}_{12} & & & \\ \mathcal{C}_{12} & \mathcal{C}_{11} & \mathcal{C}_{12} & & & \\ \mathcal{C}_{12} & \mathcal{C}_{12} & \mathcal{C}_{11} & & & \\ & & & \mathcal{C}_{44} & & \\ & & & & \mathcal{C}_{44} & \\ & & & & & \mathcal{C}_{44} \end{pmatrix}$$

As a result, a set of three equations is needed to determine all the constants. This means that three types of strain must be applied to the starting crystal. To achieve that, and in order to save computation time, we have chosen three highly symetrical types of deformation:

- The first type involves calculating the bulk modulus (K), which is related to the elastic constants by $K = \frac{1}{3}.(C_{11} + 2C_{12})$. The package *elast* calculates K by computing the curve $E_{tot}(V)$ for several user-supplied values of V. The curve is then fitted with the third-order Birch-Murnaghan [1] equation of state ($E_{tot} = a + b.V^{-2/3} + c.V^{-4/3} + d.V^{-2}$). K is finally derived using:

$$K(V_0) = \frac{1}{3}.(C_{11} + 2C_{12}) = V_0.\left(\frac{\partial^2 E_{tot}}{\partial V^2}\right)(V_0) \qquad (4)$$

- The second type involves performing volume-conservative tetragonal strains. We vary the ratio $c/a = (1 + e)$ for several user-supplied values of e leading to the strain tensor:

$$\bar{\epsilon} = \begin{pmatrix} \epsilon_1 & & \\ & \epsilon_1 & \\ & & \frac{1}{(1+\epsilon_1)^2} - 1 \end{pmatrix} \text{ or in Voigt notation: } \begin{pmatrix} \epsilon_1 \\ \epsilon_1 \\ \frac{1}{(1+\epsilon_1)^2} - 1 \\ 0 \\ 0 \\ 0 \end{pmatrix},$$

where $\epsilon_1 = (1 + e)^{-1/3} - 1$. From equation (2), we then derive, to second order in ϵ_1:

$$\frac{\phi_{tetra}}{V_0} = 3.(C_{11} - C_{12}).\epsilon_1^2 + o(\epsilon_1^3)$$

In practice, $E_{tot}(\epsilon_1)$ is fitted to a polynom P, of degree N. Thes N is limited by the number of structure changes ($N \leq$ (Number of data)-1).

Within that limitation, N is optimized to give the better least-square fit. The value of $(C_{11}-C_{12})$ is then simply computed from the value of the second derivative of P:

$$P''(\epsilon_1=0)=6.V_0.(C_{11}-C_{12}) \tag{5}$$

- Finally, the last type of deformation we have chosen is rhombohedral distortion. For that kind of strain, we vary the length of the great diagonal of the cubic cell. First, we have to define the rhombohedral unit cell vectors, $\overleftrightarrow{a}_r$, with respect to the initial cubic vectors, $\overleftrightarrow{a}_0$. We have:

$$\begin{aligned}\begin{pmatrix}\vec{a}_r\\ \vec{b}_r\\ \vec{c}_r\end{pmatrix} &= \begin{pmatrix}0 & 1/2 & 1/2\\ 1/2 & 0 & 1/2\\ 1/2 & 1/2 & 0\end{pmatrix}.\begin{pmatrix}\vec{a}_0\\ \vec{b}_0\\ \vec{c}_0\end{pmatrix} \quad \text{for face-centered lattices}\\ &= \begin{pmatrix}-1/2 & 1/2 & 1/2\\ 1/2 & -1/2 & 1/2\\ 1/2 & 1/2 & -1/2\end{pmatrix}.\begin{pmatrix}\vec{a}_0\\ \vec{b}_0\\ \vec{c}_0\end{pmatrix} \quad \text{for body-centered lattices}\end{aligned}$$

and for primitive cubic lattices, $\overleftrightarrow{a}_r=\overleftrightarrow{a}_0$. In fact, the resulting rhombohedral lattice is defined by using its related hexagonal vectors, $\overleftrightarrow{a}_H$:

$$\begin{pmatrix}\vec{a}_H\\ \vec{b}_H\\ \vec{c}_H\end{pmatrix}=\begin{pmatrix}1 & -1 & 0\\ 0 & 1 & -1\\ 1 & 1 & 1\end{pmatrix}.\begin{pmatrix}\vec{a}_r\\ \vec{b}_r\\ \vec{c}_r\end{pmatrix}$$

Then, we vary $\vec{c}_H=\vec{c}_{H0}.(1+e)$ by using several user-supplied values for e. For all cubic lattices, the resulting strain tensor is:

$$\overline{\epsilon}=\frac{e}{3}\begin{pmatrix}1 & 1 & 1\\ 1 & 1 & 1\\ 1 & 1 & 1\end{pmatrix},$$

and the corresponding elastic energy:

$$\frac{\phi_{tetra}}{V_0}=\frac{e^2}{3}.(\frac{C_{11}+2C_{12}}{2}+2C_{44})+o(e^3)$$

To extract elastic parameters, we proceed as for tetragonal strain. $E_{tot}(e)$ is fitted to a polynom, P, and we finally obtain:

$$P''(e=0)=\frac{1}{3}.(C_{11}+2C_{12}+4C_{44}) \tag{6}$$

Equations (4), (5) and (6) form the set of equations needed to determine the full elastic tensor.

3 Scripts and programs.

Three scripts drive the program flow:

- *init_elast*
  This script prepares the whole calculation. It must be run in a directory that contains valid *'case.struct'* and *'case.inst'* files. First, *init_elast* creates the following directories and sub-directories: ./elast (the main directory), ./elast/eos (directory where the calculations leading to K will take place), ./elast/tetra (tetragonal distortion calculations directory), ./elast/rhomb (rhombohedral distortion calculations directory) and ./elast/result where all the calculation results will be stored. The file *'case.struct'* is taken as the initial unstrained state. This file is stored as *'init.struct'* in the ./elast directory. Then, the template *'struct'* files are generated (./elast/eos.templ, ./elast/tetra.templ, ./elast/rhomb.templ), and the script *'init_lapw'* is launched in each of the calculation directories.

- *elast_setup*
  This script must be run in the created previously *elast* directory. The script generates all the input files for the calculation using *init.struct* and the **.templ* files. The **.struct* files thus created are all stored in the *elast* directory. *elast_setup* can be run several times to change the number and the type of structure changes. To change the volume of the unstrained state, remove all the elast/*.struct files, move the content of elast/result elsewhere, set the proper elast/init.struct and re-run *elast_setup*. Finally, *elast_setup* provides three scripts (eos.job, rhomb.job and tetra.job) that make WIEN calculate the entire set of structure changes automatically. These scripts must be adapted to your needs before running them.

- *ana_elast*
  Once the calculations are done, all the results are normally stored in the *elast/result* directory. Go to that directory and run *ana_elast*. This script will analyze all the calculated total energies according to equations (4), (5) and (6), extract the elastic constants and pressure, plot the results, and generate number of output files stored in the *result/outputs* directory.

To validate this procedure, we provide results obtained for MgO-B1 at zero pressure using LDA functional. From these low precision calculations, we obtained:

$$
\begin{aligned}
\mathrm{P} &= 0.018 \text{ GPa at volume} = 121.80976 \text{ a.u. per formula} \\
K &= 171.772\mathrm{GPa} \\
C_{11} &= 329.088\mathrm{GPa} \\
C_{12} &= 93.114\mathrm{GPa} \\
C_{44} &= 150.882\mathrm{GPa}
\end{aligned}
$$

These results differ by 10% from the computed values given in reference [2]. Other results reviewed in [2] are in better agreement.

References

[1] Birch, F. *The effect of pressure upon the elastic parameters of isotropic solids, according to Murnaghan's theory of finite strain.* J. Appl. Phys. **9**, 279-88, 1938.

[2] B.B. Karki, L. Stixrude, S.J. Clark, M.C. Warren, G.J. Ackland, and J. Crain *Structure and elasticity of MgO at high pressure.* American Mineralogist **82**, 51-60, 1997.